

**APPLICATION
FOR
UNITED STATES LETTERS PATENT**

APPLICANT NAME: John T. McDonald, et al.

TITLE: DEVELOPMENT SHELL

DOCKET NO. END920000076US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C., 20231 as "Express Mail Post Office to Addressee" Mailing Label No. EK758292920US on 03/20/01

Name of person mailing paper Christine E. Lang

Signature

Christine E. Lang

Date

3/20/01

DEVELOPMENT SHELL

5

Background Of The Invention

Field of the Invention

- 10 This invention generally relates to managing processes, and more specifically, the invention relates to methods and systems that provide infrastructures for managing processes.

Prior Art

15

Business applications such as Lotus Notes applications and Web applications that are interactive and collaborative are very difficult to design, code, roll-out and maintain because the businesses using them are changing as fast as the applications are built, tested and deployed. With today's dynamic customer environments, there is a great
20 deal of cost associated with maintaining, reworking, and deploying almost daily changes in support of the customer's changing business needs.

- Customers need their applications to expand dynamically in support of their ever growing customer sets on a global basis. With this comes the requirement to
25 customize existing application workflow, defined roles, application scope and process definitions without additional application development that can impact existing customer sets. Business and application owners need centralized administrative methods for controlling their business processes in this highly distributed application environment.

30

Summary Of The Invention

An object of this invention is to provide an architecture and a methodology that provide a highly flexible infrastructure for managing any process.

5

Another object of the present invention is to provide a development shell that addresses the need for application owners to improve the efficiency and effectiveness of their employees through control, communication and collaboration.

- 10 A further object of this invention is to allow businesses to define current business processes while remaining flexible enough to integrate new processes dynamically as the business changes.

These and other objectives are attained with a method and system for providing a
15 flexible infrastructure for managing a process. The method comprises the steps of defining a workflow type, building a workflow pointer that defines the location of a database within a network, and coding subforms for the defined workflow type. Role documents are defined and customized, process flow documents are built for the defined workflow type, and action control documents are built for determining what
20 happens to the defined workflow type.

Preferably, the present invention is defined on a process model that makes it generic enough to be a scaleable architecture. The workflow model is a block model for diagramming any process. The basic model structure is based on arrows that define
25 process inputs, outputs, controls and supports. With this basic building block, it is possible to sit with a customer and draw any business process so that it can be understood by all members of any team. With this model, both customers and providers of services can focus on any selected process box and come to agreement on the inputs and outputs of that step. The invention takes this principal and integrates it
30 into the customer's application as a usable solution that can be implemented and/or changed as fast as it can be agreed upon.

The development shell architecture of this invention is based on the concept that many business processes are interrelated and need to communicate and share information, to have their own workflow, and to be able to integrate workflow with other application workflows and processes. These business processes also need to help owners define their processes, create new ones and integrate others, and to improve the efficiency and effectiveness of end users through control, communication and collaboration. In addition, the present invention provides a long term solution with a rapid, low cost implementation, and is open, flexible and scaleable with a building block design techniques. Also, the method and system of this invention can be administratively maintained through a single point of control, provides major development and application maintenance life-cycle savings, and provides a common methodology.

Further benefits and advantages of the invention will become apparent from a consideration of the following detailed description, given with reference to the accompanying drawings, which specify and show preferred embodiments of the invention.

Brief Description Of The Drawings

Figure 1 diagrammatically illustrates a general workflow paradigm.

Figure 2 generally outlines a method embodying this invention.

Figure 3 schematically depicts database templates used in the preferred implementation of the present invention.

Figure 4 illustrates a procedure for customization of a workflow.

Figure 5 shows a work flow feature.

Figure 6 illustrates the processing of an action control document.

Detailed Description Of The Preferred Embodiments

5 The design and functionality of the development shell of this invention is predicated on the general workflow paradigm represented in Figure 1. In this paradigm, a workflow document is created by a submitter requesting work. The document contains the information necessary to perform that work. The document is reviewed, negotiated and/or accepted and assigned to someone to perform the work, or the
10 request may be rejected or backlogged. The work is performed and results documented within the workflow document or in sub-documents related to the workflow document. These sub-documents may, in turn, be related to other workflow documents. A review and acceptance process may be performed and the result recorded in the workflow document.

15 The ability to create new workflow documents from an existing document, update any documents along a chain of related documents so as to track the status of particular work effort, and the ability to navigate among any and all related documents at any point in the workflow is necessary for efficient and effective completion of the
20 workflow.

The above paradigm is a generic process that could be applied to many business processes. The development shell of this invention has been developed with this in mind so that application solutions are flexible and easily adapted to any business
25 process.

Figure 2 outlines a preferred embodiment of the present invention. With this preferred method, at steps 12, 14 and 16, a workflow type is defined, a workflow pointer is built that defines the location of a database within a network, and subforms are coded for
30 the defined workflow type. At step 20, role documents are defined and customized; at step 22, process flow documents are built for the defined workflow type; and at step

24, action control documents are built for determining what happens to the defined workflow type.

Preferably, with reference to Figure 3, the development shell is a set of four Notes database templates. The templates are Personnel 30 (for applying names to an application), History 32 (for recording who is doing what), Administration 34 (for storing all application customizations for keywords, roles, documentation, help, actions and controls) and the Application template 36 (for maintaining and controlling the application interface). These templates may be utilized by any application as the basic foundation (architecture) for developing any workflow solution. By applying the development shell templates, an application may make use of defined features within the development shell through a process called customization. Customization means an application workflow process can be defined by building a set of documents that control how the application's workflow will function without writing any code. Coding is centered around defining the customer's application interface (subforms, dialogs, views, navigators and agents).

With reference to Figures 3 and 4, customization of a workflow is separated into a series of steps. At step 52, a workflow is defined. A set of keyword documents 54 is built into the administration database template 36 that uniquely identify the workflow type. These documents become the keys used throughout the application to control how the application interface works. The application interface, consisting of application defined and developed views, subforms, events, actions and agents, are all managed by the defined workflow type.

At step 56, a workflow pointer 60 is built. This is an action document that is built into the application database 40 and defines the location of the application databases within the network. These pointers are used when various development shell customized features are invoked to determine where to find the various databases for lookups, workflow processing, associating (linking and chaining workflow documents), posting (updating linked and chained documents), writing History 34 (another development

template) and more (field level help, integrating application documentation, integrating personnel 32 (another development shell template)).

At step 62, workflow defined subforms are coded. Development shell workflow
5 makes use of Notes subform design elements. The application developer builds the application specific subforms required for the defined workflow. The subforms are then displayed to the application users through the development shell main form based on the application defined workflow type. The development shell main form is what provides the base foundation for all workflow documents. Subforms are usually
10 broken out into specific application defined process elements. Most application workflows revolve around creating and submitting a document, reviewing the submitted request and approving the document for further processing, providing a resolution to the approved request and signing off on the resolution. This implies collaboration by multiple people playing various roles at various times within a single
15 document or across multiple related workflow document types.

Step 64 is to define and customize role documents. The developer builds a set of role definition keywords and role documents that control who does what within a workflow type or set of workflow types. These role documents allow the application
20 to remain generic and control when a name is applied to a specific workflow document. The name is linked to the role and not the workflow document, reducing the cost of maintaining and controlling who can do what within a specific workflow type and when they can do it. A workflow type can be rolled out to multiple varied groups and organizations by building additional role documents in support of them. It
25 is these role documents, when applied to an application, that allow the separation of responsibility within a single application instance without the need for utilizing reader and author name fields (these fields drastically reduce the performance of any database).

30 Step 66 is to build the workflow types process flow documents. A set of customized documents called status actions 70 are built within the administration database

template 36. These status action documents are used by the defined workflow type to determine what action a given user may perform within a given workflow document. Control is based on the workflow type, the current status of the workflow document and the role or document fields containing the user's name. Actions are invoked when
5 a user presses the default workflow action button or through application defined fields, buttons, events and/or agents.

Step 72 is to build the actions control documents 74. A set of action control documents that determine what happens to a defined workflow type is built in the
10 administration database template 36. The action control document run is defined by the status action document. When a user takes a particular status action, the defined action control document is processed by the development shell. The action control document defines what fields are validated (applications do not need to code this in fields), what pre and post confirmations are presented, what fields are updated (within
15 the workflow type or another defined workflow type) and what mail is to be sent and to whom. Field updates include, but are not limited to, stuffing a value from one field into another field, looking up data and presenting it to the user, stuffing looked up roles into fields, arithmetic and if-then-else logic, posting, getting and posting, executing most notes functions, displaying application defined dialogs, and building
20 up or downstream documents. Many other types of field updates may also be used in the practice of this invention.

An important aspect of the invention is its ability to provide generic workflow features and the ability to customize them, along with the type and variety of functions. Any
25 application workflow can be defined, managed and controlled through a generic set of development shell agents that manage and control an application. The customized action and control documents are generic enough that they can be built to control how an application will process. This solution encompasses security, control, collaboration, communication and connectivity for Notes and Domino (Web)
30 applications through a centrally administered single-point-of-control. Because the

workflow is managed in the administration database 36 as documents, the applications workflow can literally be modified without changing any code.

- Basic components of the development shell workflow features are illustrated in Figure 5. As represented at 80, users of a development shell defined application interface with the workflow feature through application defined events, buttons or by utilizing the default development shell Work-Flow Actions button. The development shell Determine Actions Agent 82 gains control and queries the Administration databases Status Actions View 84 to determine what actions the user is authorized to take.
- Authorization is based on the workflow document type, the current status of the workflow document and the authorizations allowed for the selected actions. The result of the query is a list of authorized actions for that user.

- If Determine Actions was invoked as auto-run, then control is routed to the Take Action Agent 86 to process the action control document associated with the application defined auto-run event. Otherwise, the development shell default dialog 90 is displayed and the user is presented with a list of actions they may pick. Once the user selects an action, control is routed to the Take Action Agent 86. This agent processes the selected action control document and returns control to the end user.

- Figure 6 shows the high level inner working principals of how the development shell Take Action Agent processes an Action Control Document 90 that was selected by an end user for any given Action. The Action Control Document contains the customized data from an application for a given action. These data are processed by the Take Action Agent, and control is returned to the end user when the agent completes. Various functions that any given Action Control Document might do (as defined by the application) are: workflow document field validations; pre and post confirmation messages; creation of new workflow documents; field updates into the current workflow document or other associated workflow documents within the application or cross applications and servers, if so authorized; building of a mail memo that would be

sent on the workflow document save event; and processing of additional Action Control Documents with optional branching logic.

5 The preferred embodiment of the invention, as described above in detail, has a number of important advantages. For instance, the invention may be used to achieve the client's objectives of tracking and controlling a business process. The invention may be used to improve efficiency and effectiveness by presenting a common look and feel throughout the application for workflow features and providing the ability to navigate easily between related documents. Also, this invention reduces database maintenance and development effort by utilizing administrative documents for defining keywords
10 and for controlling the workflow process, functionality, roles and more. In addition, the invention provides a generic design that is flexible and can be customized easily to support any business process requiring tracking and control. This invention can be used with many types of business architectures, including Lotus Notes, Domino, and
15 Web platforms.

While it is apparent that the invention herein disclosed is well calculated to fulfill the objects stated above, it will be appreciated that numerous modifications and embodiments may be devised by those skilled in the art, and it is intended that the
20 appended claims cover all such modifications and embodiments as fall within the true spirit and scope of the present invention.